EXHIBIT P

RSC-164/ RSC-164i Data Book

GENERAL DESCRIPTION

The RSC-164 and RSC-164i are low-cost speech recognition ICs designed for use in consumer electronics. They combine an 8-bit processor with neural-net algorithms to provide high-quality speaker-independent speech recognition, speaker-dependent speech recognition, and speaker verification. The chips also support speech synthesis, voice record/playback, 4-voice music synthesis, speaker verification, and system control. These CMOS device includes on-chip RAM, ROM, 16 general-purpose I/O lines, A/D and D/A converters, and a 4-MIPS dedicated processor.

In addition to providing the horsepower needed to perform speech recognition and speech synthesis, the processor has sufficient cycles available for general-purpose product control. The RSC Development Kit allows developers to create custom applications for the RSC chips. The Development Kit includes an assembler, linker, simulator, in-circuit emulator, and library of Sensory technology object code.

The highly-integrated nature of these chips reduce external parts count. A complete system may be built with few additional parts other than a battery, speaker, microphone, and audio input support circuitry. Low power requirements make the RSC chips an ideal solution for battery-powered and hand-held devices.

The RSC chips use a pre-trained neural network to perform speaker-independent speech recognition, while high-quality speech synthesis is achieved using a time-domain compression scheme that improves on conventional ADPCM. Four-voice music synthesis allows multiple, simultaneous instruments for harmonizing. Dynamic AGC control can compensate for people not optimally positioned with respect to the microphone or for people who speak too softly or loudly.

FEATURES

High-Performance Processor

- 4-MIPS performance at 14.32 MHz
- 12-16 general-purpose I/O lines
- Interrupts, timers and counters
- Fully static operation; clock speed from DC to 14.32 MHz

Highly-Integrated Single-Chip Solution

- Internal 64K ROM
- 384 bytes RAM
- 12-bit A/D (Analog to Digital converter)
- Pulse width modulator for direct speaker drive

Low Power Requirements

- Requires single 3.5 to 5.0 volt supply.
- Typical current drain is 10 mA
- Low-power 32,768 Hz oscillator for clock applications

High-Quality Recognition and Synthesis

- Recognition accuracy better than 96% (Speaker Independent) and 99% (Speaker Dependent).
- Synthesis data rates from 5,000-15,000 bits per second
- 4-voice music synthesis capabilities
- Optional AGC control compensates for variations in input signal

Easily Expanded to larger-scale systems

- Separate 16-bit Address and 8-bit Data buses compatible with common memory components
- Separate Code and Data address spaces and memory strobes
- Instruction set compatibility between RSC-164 and RSC-164i

IMPORTANT NOTICES

Sensory reserves the right to make changes to or to discontinue any product or service identified in this publication at any time without notice in order to improve design and supply the best possible product. Sensory does not assume responsibility for use of any circuitry other than circuitry entirely embodied in a Sensory product. Information contained herein is provided gratuitously and without liability to any user.

Reasonable efforts have been made to verify the accuracy of this information but no guarantee whatsoever is given as to the accuracy or as to its applicability to particular uses.

Applications described in this data sheet are for illustrative purposes only, and Sensory makes no warranties or representations that the Interactive Speech™ line of products will be suitable for such applications. In every instance, it must be the responsibility of the user to determine the suitability of the products for each application. Sensory products are not authorized for use as critical components in life support devices or systems.

Sensory conveys no license or title, either expressed or implied, under any patent, copyright, or mask work right to the Interactive Speech™ line of products, and Sensory makes no warranties or representations that the Interactive Speech™ line of products are free from patent, copyright, or mask work right infringement, unless otherwise specified. Nothing contained herein shall be construed as a recommendation to use any product in violation of existing patents or other rights of third parties. The sale of any Sensory product is subject to all Sensory Terms and Conditions of Sales and Sales Policies.

Data Book Revision 2.5

August 1996

SENSORY

521 East Weddell Drive
Sunnyvale, CA 94089

TEL: (408) 744-9000
FAX: (408) 744-1299

P/N 80-0007-5

Table of Contents

1. INTRODUCTION **1**

2. RSC-164 ARCHITECTURE **2**

3. DIFFERENCES BETWEEN THE RSC-164 AND THE RSC-164i **3**

4. MEMORY ORGANIZATION **4**

5. MEMORY MAP **5**

6. GENERAL PURPOSE I/O **6**

7. INTERRUPTS **8**

8. RESET AND CLOCKS **10**

9. TIMERS AND COUNTERS **11**

10. ANALOG OUTPUTS **11**

11. HARDWARE DEBUG FEATURES **12**

12. DESIGN CONSIDERATIONS **13**

13. OMNI-DIRECTIONAL MICROPHONE **17**

14. POWER CONSUMPTION AND POWER SUPPLY CONSIDERATIONS **19**

15. DIE BOND PAD, PLCC AND QFP PIN DESCRIPTIONS (VERSION "B") **20**

16. ABSOLUTE MAXIMUM RATINGS **21**

17. D.C. CHARACTERISTICS **21**

18. A.C. CHARACTERISTICS *(*EXTERNAL MEMORY ACCESSES*)* **22**

19. RSC-164 INSTRUCTION SET **23**

20. RSC-164 SPECIAL FUNCTION REGISTER (SFR) SUMMARY **26**

21. SPECIAL DATA SPACE ADDRESSES SUMMARY **40**

22. ORDERING INFORMATION **42**

1. Introduction

The RSC-164 is the first in a family of high-performance 8-bit microprocessors featuring a high level of integration, targeted to high-accuracy, low-cost speech recognition applications. The RSC-164 family is designed to bring a high degree of integration and versatility to low-cost, power-sensitive consumer applications.

Various functional units have been integrated onto the CPU core in order to reduce total system cost, yet increase system reliability, without degrading system performance. By integrating data conversion, recognition and synthesis functionality, and ROM[1] storage with a CPU core on a single chip, dramatic cost and power reductions are achieved. Thus, the RSC-164 family is able to provide 4 MIPS of integer performance at 14.32 MHz. This allows customer applications to achieve maximum performance at minimum cost.

The CPU core embedded in the RSC-164 is an 8-bit, variable-length-instruction, microprocessor. The instruction set is loosely based on Intel's 8051™, having a variety of addressing mode *mov* instructions. But the RSC-164 processor avoids the limitations of dedicated A, B, and DPTR registers by having completely symmetrical source and destinations for all instructions. The 384 bytes of internal RAM are organized as a Register Space. All arithmetic operation instructions may be applied to any register. Any pair of adjacent registers (at an even address) may be used as the 16-bit pointer to either the source or destination for a data movement instruction. Instruction classes allow the pointer to access internal or external Code Space, internal Register Space, or external Data Space.

Architecturally, the RSC-164's separate data and address buses allow use of standard EPROMs, ROMs, and SRAMs with little or no additional decoding. Provision of separate read and write signals for each external memory space further simplifies interfacing.

Creating applications using the RSC-164 or the RSC-164i requires the development of electronic circuitry, software code, and speech/music data files ("linguistics"). This document provides detailed information on those aspects of the RSC-164 architecture that are important to product designers and programmers. It describes the physical interface to the chip, printed circuit board layout and other design considerations, the RSC-164's instruction set, and memory organization. Refer to the RSC Development Kit Manual for information on using Sensory's technology code for speech recognition, speaker verification, speech synthesis, and voice record and playback. Description of vocabulary development ("linguistics") information is beyond the scope of this document and is covered in Design Note 1 (P/N 80-0014-1).

[1] The RSC-164 internal ROM contains primarily Sensory library code; The internal ROM in the RSC-164i is application specific, with the amount available for user applications decreasing as the number of synthesis words or other technology usage increases.

2. RSC-164 Architecture

The RSC-164 is a highly integrated device that combines:

- An 8-bit RISC microprocessor.
- On-chip ROM (64 Kbytes) and RAM (384 bytes), and the ability to address off-chip RAM or ROM.
- An analog-to-digital converter, a digital-to-analog converter, and a pulse width modulator.

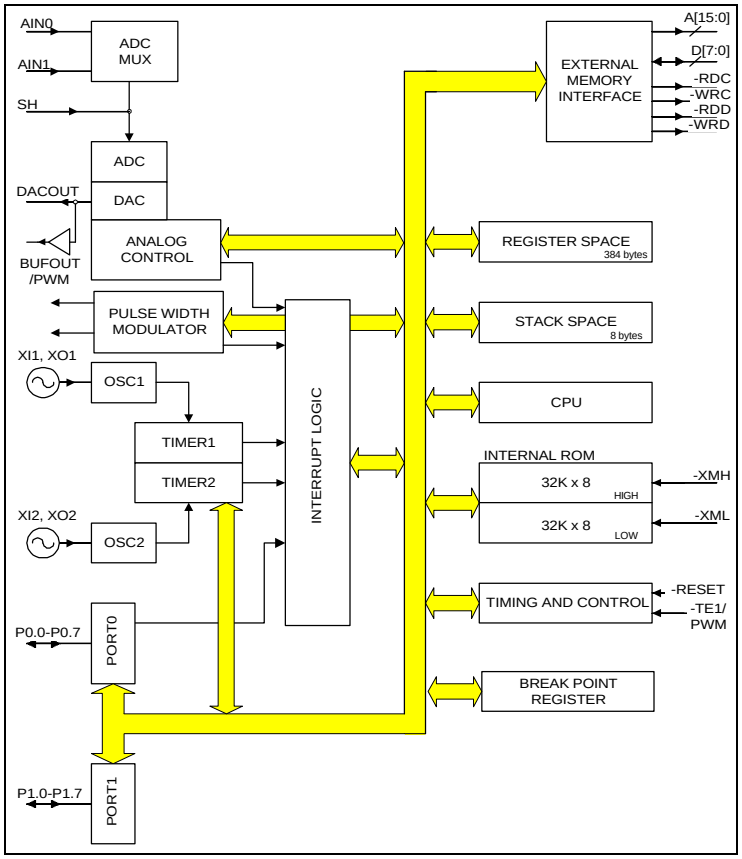


Figure 1 -- RSC-164 Block Diagram

The RSC-164 has an external memory interface for accessing external RAMs or ROMs. It also has an internal ROM that can be enabled or disabled (partially or fully) by pin inputs (signals -XMH, -XML; see page 4). The RSC-164i does not access external parallel memory and relies solely on the internal ROM for its program storage needs.

The 8-bit processor can directly access 384 on-chip registers (RAM), of which 352 are general purpose registers and 32 are Special Functions Registers (SFRs). The instruction set accessing these registers is completely symmetrical, allowing *mov*s, arithmetic, and logical operations with any register as the destination. Two bi-directional ports provide 16 general purpose I/O pins to communicate with external devices (see page 6). The RSC-164 has a high frequency (14.32 MHz) oscillator as well as a low frequency (32,768 Hz) oscillator suitable for timekeeping applications. The processor clock can be selected from either source, with a selectable divider value. Sensory's technology code requires the use of the 14.32 MHz clock. The device performs speech recognition when running at 14.32 MHz, with an optional divide-by-2 CPU clock (see page 10).There are two programmable 8-bit counters / timers, one derived from each oscillator.

A microphone with an external preamp converts sound into two audio signals that are fed into the AIN0 and Ain1 inputs of the RSC-164. The gain of the external preamp may be controlled by the RSC-164 in some designs. The RSC-164 uses an ADC (Analog-to-Digital Converter) and a Sample and Hold (SH) circuit to convert the incoming analog speech signal into digital data. The output audio signal of the RSC-164 is derived either from a DAC (Digital-to-Analog Converter) or a PWM (Pulse Width Modulator).

In addition to its on-chip ROM and RAM, the RSC-164 has 8 data lines (D[7:0]) and 16 address lines (A[15:0]), along with associated control signals (-RDC, -RDD, -WRC, -WRD, -XML, -XMH) for interfacing to external memory. The memory control signals on the RSC-164 and the processor instruction set provide independent Code and Data spaces, allowing configuration of systems up to 192 Kbytes with no additional hardware decoding. The RSC-164i does not contain these address, data, and control signals. The RSC-164 features 16 general-purpose I/O pins (Px.y) for product and memory bank control; the RSC-164i has 12 general purpose I/O lines.

3. Differences Between the RSC-164 and the RSC-164i

The main difference between the RSC-164 and the RSC-164i is that the former can access external memory devices. Also, the RSC-164 has four more general purpose I/O pins than the RSC-164i. The instruction sets for both devices are identical[2]. Although the RSC-164 provides significant and flexible expansion capabilities through the use of external RAM or ROM, the RSC-164i must rely on limited internal memory for all of its ROM and RAM requirements. The RSC-164i can also interface to serial memory through the I/O lines for data storage. These finite resources restrict the capabilities of products based on the RSC-164i. The product specification for an RSC-164i must be carefully crafted in consultation with Sensory to maximize the use of on-chip memory. Each application will have its own specific limitations, but the table below summarizes some useful guidelines for planning purposes. Not all of the maximums can be achieved in a single RSC-164i design. For example, a recognition vocabulary of 40 words may limit the speech synthesis to substantially less than 25 seconds.

Description	RSC-164	RSC-164i
Capabilities:		
Speaker independent (SI) recognition	✓	✓
Speaker dependent (SD) recognition	✓	✓
Speech synthesis and special sound effects	✓	✓
Speaker verification	✓	✓
Four-voice music generation	✓	**limited support**
Voice record and playback	✓	**not supported**
SI Recognition Capacity :		
Maximum number of words per recognition set[3]	**14**	**14**
Total recognition vocabulary size in words, all sets	**unlimited**	**40 words[4]**
SD Recognition Capacity :		
Maximum number of words per recognition set[3]	**64**	**64[5]**
Total recognition vocabulary size in words, all sets	**unlimited**	**512[5]**
Speaker Verification Capacity :		
Number of speakers identified per set[3]	**64**	**64[5]**
Synthesized Speech Capacity:		
Maximum total length of all messages	**unlimited**	**25 seconds[4]**
Music Synthesis Capacity		
Number of simultaneous independent musical voices	**4**	**4**
Number of musical octaves available	**2-4[6]**	**2**
Number of musical tunes available	**unlimited**	**6**
Requirement for custom ROM masks:		
Use of the RSC-164i requires custom-masked ROMs. Custom-masked parts are not stocked by Sensory	**Custom masked ROM not required**	**Custom masked ROM required**

[2] Software for RSC-164i applications may be completely developed and verified using the RSC Development Kit and an external 64K ROM memory before committing to an RSC-164i ROM mask.
[3] Practical limitations to maintain accuracy above 95%.
[4] Assumes the use of on-chip ROM only.
[5] Assumes the use of external serial memory devices.
[6] Depends on choice of musical instrument.

4. Memory Organization

Internal Memory

Internal ROM is organized as two banks of 32K bytes each, both mapped into code space. Either of the two internal banks may be independently disabled using external inputs; input pin -XML disables the lower 32K [0000h-7FFFh] bank, while input pin -XMH disables the upper 32K [8000h-FFFFh] bank. When a bank is disabled, read accesses to it are directed to off-chip code space. Write accesses to the code space are directed to external memory off-chip. Except for specific addresses in the last page of memory (described on page 5), read and write accesses to data space are always directed to external memory.

External Memory

The RSC-164 allows for extended message lengths and expanded program functionality by using external memory. There are 30 pins that provide an interface between the RSC-164 and external ROM or RAM. The 16 address line outputs, A[15:0], are shared for accesses to external code space or data space. The 8 data lines, D[7:0], are bi-directional, and are normally inputs except when there is a write to external memory. Refer to *MEMORY MAP* (on page 5) for details on accessing external code and data spaces through *movc* and *movx* instructions.

The RSC-164 uses the -RDC, -WRC, -RDD and -WRD signals to strobe data to or from memory or I/O devices. The -RDC and -WRC strobes are provided for accessing code space, while the -RDD and -WRD strobes are used to access data space. These four memory strobes are all active low (see page 22 for timing information). Using these strobes, the RSC-164 can directly access 64K of external code space and 64K of external data space in addition to its internal 64K of code space. External memory and I/O devices can reside in either code space or data space, as determined by the user application. Executable code *must* reside in code space; tables and other data may reside in code space or data space. Using I/O bits (from Port 1), additional external decoding can be used to bank select between multiple RAMs or ROMs. This method allows for external storage requirements larger than the combined 192K addressed directly by the RSC-164.

The RSC-164i does not address external memory, and does not include the address A[15:0], data D[7:0], or control[-RDC, -WRC, -RDD, -WRD, -XML, -XMH] signals for accessing external memory.

Use of 100nS or faster external memories is recommended when operating at 14.32 MHz with one wait state. Given below is a simple representation of the internal ROM memory control and a block diagram of an external memory implementation illustrating a single 32Kx8 data space RAM and a bank switched 1 megabyte code ROM. The P1.4 signals controls whether -RDD accesses RAM or ROM, and P1.0-P1.3 provide the bank address bits.

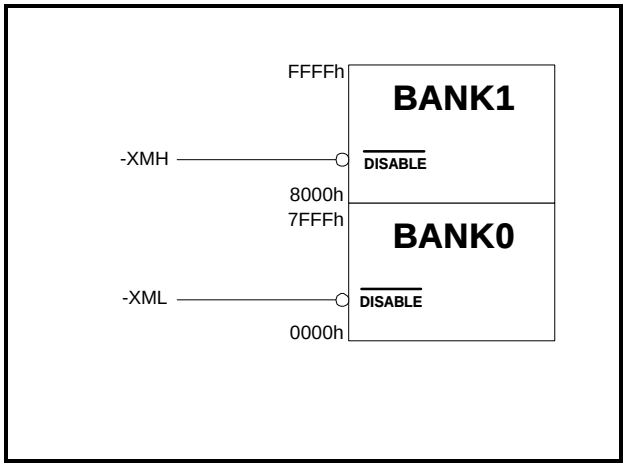


Figure 2-- Internal ROM memory control

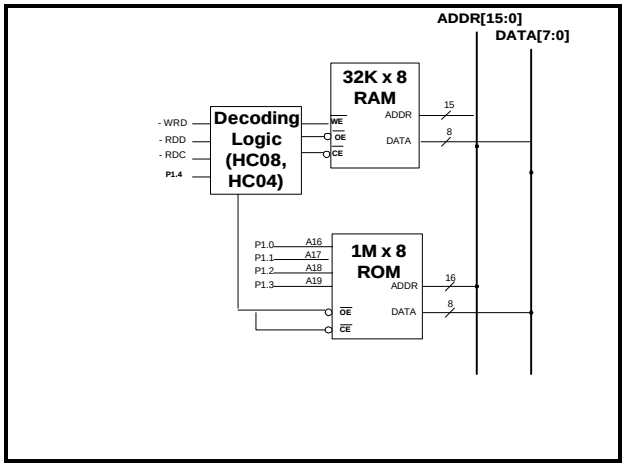


Figure 3 --External memory implementation

5. Memory Map

The RSC-164 has three address spaces: Code Space, Data Space, and Register Space. Code space is typically ROM. Data space may be either ROM or RAM. Register space is limited to on-chip SRAM. The instruction set provides separate instructions for accessing each space. Executable code *must* reside in code space; tables and other data may reside in code space or data space. Register space is intended primarily for variables.

The internal ROM and off-chip code space memory may be accessed using *movc* instructions. The off-chip data space is accessed using *movx* instructions, while the on-chip register space is accessed using *mov* instructions. For the RSC-164, writes to code space are always directed off-chip, while the RSC-164i ignores writes to code space.

The internal ROM is organized as two code space banks of 32K bytes each. The RSC-164 allows the banks to be independently disabled using external inputs: the low bank (address range 0000h-7FFFh) can be disabled by asserting pin input -XML while the high bank (address range 8000h-FFFFh) can be disabled by asserting pin input -XMH. This feature is used extensively to expand addressable code space beyond 64K bytes. The RSC-164i relies entirely on the internal ROM for its storage needs and therefore the internal ROM cannot be disabled.

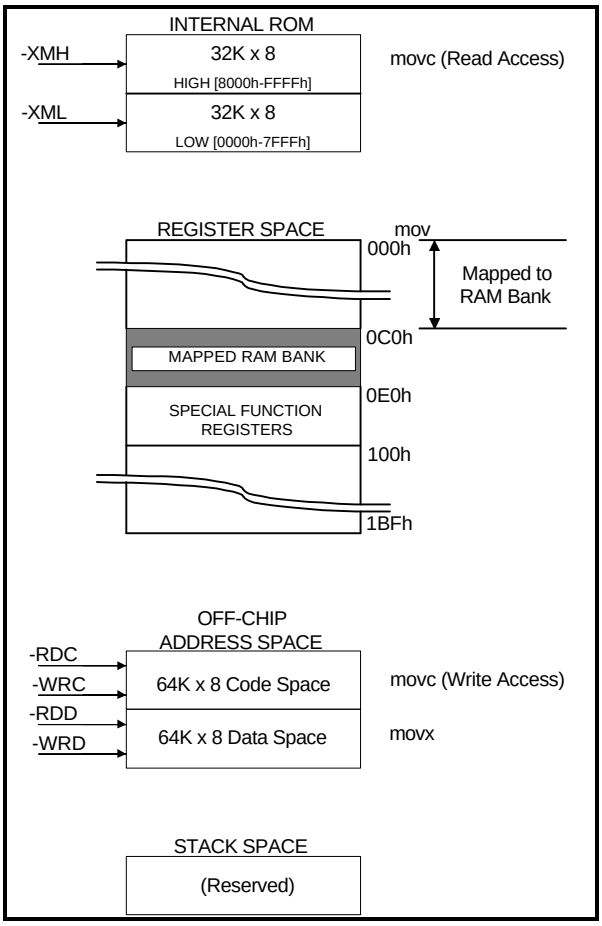


Figure 4 -- Memory Map

The SRAM register space supports 8-bit addresses, so only 256 bytes may be directly addressed. General purpose registers are located between addresses 000h and 0BFh. A 32-byte bank of SFRs (special function registers) resides at addresses 0E0h-0FFh. The 32-byte bank at addresses 0C0h-0DFh may be mapped to any of the six lower 32-byte banks in page 0 (addresses 000h through 0BFh), or to six additional 32-byte banks in page 1 (addresses 100h-1BFh. A special function register controls this mapping, providing a total of 384 bytes of SRAM register space.

Off-chip memory (and memory-mapped I/O) is accessed using a 16-bit address bus and an 8-bit data bus. Separate read / write strobes are generated for access to external code and data spaces. This allows the RSC-164 to directly access 64K bytes of external code memory and 64K bytes of external data memory. Bank switching is commonly implemented using I/O pins to select additional off-chip memory. Because the RSC-164i lacks the ability to access external parallel memory, it does not have the external address and data buses and the associated read and write strobe outputs.

Certain addresses in the range of 0FF00h-0FFFFh of Data Space are mapped internally, so addresses in this last page of data space are not generally accessible in external memory. See Special Data Space Addresses, page 40.

The RSC-164 allows software to adjust the speed of off-chip memory access. This allows using fast memory for performance needs or (if feasible) slower memory for cost savings. The off-chip memory access time can be stretched using wait states defined by the BANK register, and the software can dynamically change the wait state value depending on the particular memory or I/O peripheral.

There is minimal stack space on chip. The stack is required for interrupts and allows a very limited number of nested calls. Programmers are encouraged to write inline code instead of making extensive subroutine calls. The use of macros simplifies generation of inline code. Sensory's technology code makes extensive use of a software stack to allow more deeply nested calls. Macros using this software stack are accessible to developers.

6. General Purpose I/O

The RSC-164 has 16 general purpose I/O pins (P0.0-P0.7, P1.0-P1.7). Each pin can be programmed as an input with weak pull-up (~200KΩ equivalent device); input with strong pull-up (~10KΩ equivalent device); input without pull-up, or as an output. This is accomplished by having 32 bits of configuration registers for the I/O pins (Port Control Register A and Port Control Register B for ports 0 and 1).

After reset, all of the I/O pins are set to be inputs with weak pull-ups. Designers may make use of this start-up feature to assure enabling or disabling of particular functions controlled by I/O pins.

- As outputs, these pins can source or sink 4 mA with a voltage drop of < 0.5V.
- As inputs, V_{IL} < 0.75V for all V_{DD} from 3.5 to 5.0V. V_{IH} > 2.5V over the same range.

All I/O pins are diode clamped to V_{DD} and ground, and are capable of sinking up to 200 mA if V_{DD} is exceeded.

In addition to providing general purpose I/O, port 0 bit P0.0 can serve as an interrupt input (using IMR and IRQ registers).

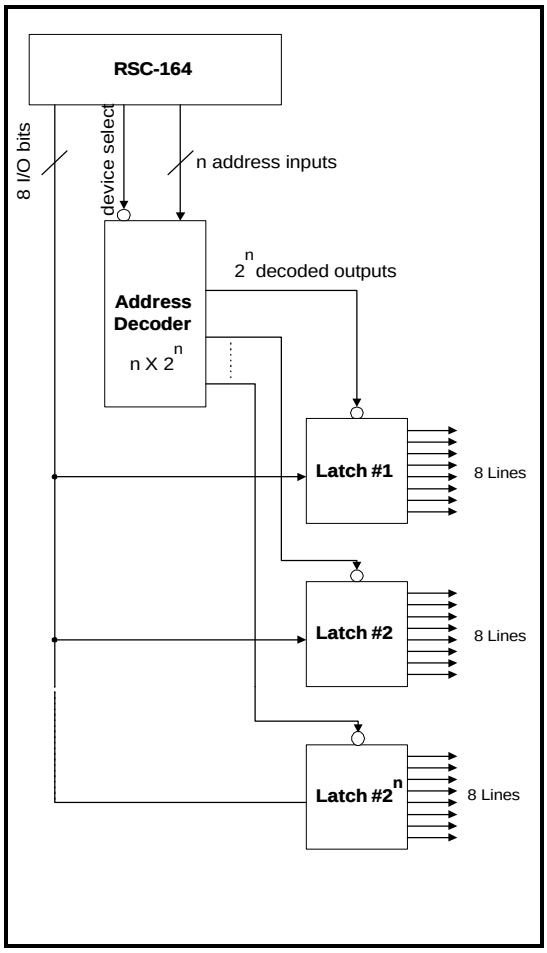


Figure 5 -- I/O Expansion

It is straightforward to add more I/O capability to the RSC-164 devices using simple logic chips. If a large number of output pins is required for a design, an address decoder can be used to select from a number of latches as shown in Figure 5.

In applications using external memories at least two (and typically more) of these I/O lines will be used for bank switching. Some of Sensory's Library Functions make specific assumptions about the configuration and operation of particular I/O pins. For example, all code space banking systems use P1.6 and P1.7 for controlling the -XML and -XMH signals. Typically some of the remaining pins on port1 are used for selecting externally banked code space ROM. This may be done by connecting port pins to Address bit 16 and higher pins of large memory devices (see Figure 3, page 4). Thus, typically Port 1 is used for memory control and Port 0 is used for general I/O functions such as keyboard scanning.

Two additional I/O lines (Port 0, bits 6 and 7) are also needed for designs requiring full AGC control of the preamplifier. However, many applications will be able to use the simpler, fixed-gain preamplifier described on page 14, which does not use these I/O lines.

7. Interrupts

The RSC-164 allows for five interrupt sources, as selected by software. Each has its own mask bit and request bit in the IMR and IRQ registers respectively. The global interrupt enable flag, which enables or disables all interrupts, is located in the FLAGS registers. Bit assignments for the IMR and IRQ registers are listed on page 38. The following events can generate interrupts:

- Positive edge on Port 0, bit 0
- Overflow of Timer 1
- Overflow of Timer 2
- Sensory Reserved functions
- Completion of PWM sample period

If an IRQ bit is set high and the corresponding IMR bit is set high and the global interrupt enable bit is set high, an interrupt will occur. Interrupts cannot be nested. The flags register is copied to a holding register and then the global interrupt enable is cleared, preventing subsequent interrupts until the IRET instruction is executed. The IRET instruction will restore the flags register from the holding register.

If the corresponding mask register bit is clear, the IRQ bit will not cause an interrupt. However, it can be polled by reading the IRQ register. IRQ bits can be cleared by writing a 0 to the corresponding bit at address 0FEh (the IRQ register). IRQ bits can *not* be set by writing to 0FEh. Writing a one is a no-op.

The IRQ bits must be cleared within the interrupt handler by an explicit write to the IRQ register rather than by an implicit interrupt acknowledge.

Important: clear interrupts this way:

```
        mov     IRQ, #BITMASK       ; right
```

not this way:

```
        and     IRQ, #BITMASK       ; wrong
```

The ‘and’ instruction is not atomic, it is a read-modify-write. If an interrupt occurs during an ‘and IRQ’ operation the interrupt will be cleared before it is seen, possibly disabling the interrupt until the system is reset.
Because you cannot set bits in the IRQ register, a ‘mov IRQ’ is a save, effective, and atomic way to clear bits in the IRQ register. Use it the way you would use an ‘and’ in other registers.

Note: if Port 0.0 (the external IRQ) is set as an *output*, the external IRQ flag will be set if the output is driven from 0 to 1 under program control.

For each interrupt, execution begins at a different code space address:

Interrupt #0	Address 4
Interrupt #1	Address 8
Interrupt #2	Address 0Ch
Interrupt #3	Address 10h
Interrupt #4	Address 14h

Normally the instruction at the interrupt address is a jump to an Interrupt Service Routine (ISR). This jump is called a *vector*. The vectors located at each of these addresses are typically in ROM. If the -XML signal is inactive (high), an internal ROM vector will be used. In this case if the vector address is in the range 0000h-7FFFh, the code will vector to internal ROM. If the vector address is in the range 8000h-0FFFFh, the code will vector either to internal ROM or external ROM depending on the setting of the -XMH signal.

If the -XML signal is active (low) when an interrupt occurs, an external ROM vector will be used. If external ROM banking is being employed, any of several different 64K banks could receive the interrupt, depending on which one is selected in the banking scheme. This illustrates the importance of paying careful attention to all possible interrupt conditions when using code space bank switching.

8. Reset and Clocks

Reset

The reset pin, -RESET, is an active low Schmitt trigger input. The reset pin is provided with hysteresis in order to facilitate power-on reset generation via an RC network. Reset is held internally for 10 msec after the -RESET input signal is deasserted. This allows the oscillator to stabilize before enabling other processor subsystems.

Oscillators

Two independent oscillators in the RSC-164 provide a high-frequency clock and a 32kHz time-keeping clock. The oscillator characteristics are as follows:

Oscillator #1	Pins XI1 and XO1	14.32 MHz (3.5V-5.0V)
Oscillator #2	Pins XI2 and XO2	32,768 Hz (3.5V-5.0V)

Oscillator #1 works with an external crystal, a ceramic resonator or LC. Use of Oscillator #2 requires a crystal for precise timekeeping.

Both oscillators have an enable control. When disabled, the inverter is high-impedance, and a weak pull-up device (~100 KÙ) holds the inverter output high. Both oscillators are controlled by the Clock Control Register (CPU register 0E8h). By default, oscillator #1 is enabled by reset, while oscillator #2 is disabled by reset. The effect of reset therefore requires that oscillator #1 be functional in all designs. The Clock Control Register also determines internal division of the CPU clock source (see below).

Each oscillator has an associated timer that is fully programmable. The RSC-164 timers are described on page 11.

Processor Clock

The RSC-164 uses a fully static core: the processor can be stopped (by removing the clock source) and restarted without causing a reset or losing contents of internal registers. Static operation is guaranteed from DC to 14.32 MHz. The processor clock is selected from either the oscillator #1 output (gated by wake-up 10 mS delay) or the oscillator #2 output, based on bit 2 of the Clock Control Register. This bit is cleared by reset, which selects oscillator #1. It is the responsibility of the firmware not to select oscillator #2 until both oscillators have been enabled and stabilized.

After source selection, the processor clock can be divided-down in order to limit power consumption. Bits 3 and 4 of the Clock Control Register determine the divisor for the processor clock. Between zero and seven wait states must also be selected for the processor clock. Wait states are inserted on reads or writes to all addresses except Register Space RAM.

Sensory technology code must run with a processor clock of 14.32 MHz, a clock divisor of one, and one wait state. This creates internal RAM cycles of 70 nsec duration and internal ROM or external cycles of 140 nsec duration. Careful design of external decoding logic and close analysis of gate delays may allow operation with memories having 120 nsec access times.

9. Timers and Counters

The two independent oscillators of the RSC-164 provide counts to two internal timers. Each of the two timers consists of an 8-bit reload value register and an 8-bit up-counter. The reload register is readable and writeable by the processor. The counter is readable with precaution taken against a counter change in the middle of a read. If the processor writes to the counter, the data is ignored. Instead, the counter is preset to the reload register value. That is, *any* write to a counter will cause it to be reloaded. This is the usual way of initializing the counter. When the timer overflows from FF to 00, a pulse is generated that sets IRQ #0 (timer #1) or IRQ #1 (timer #2). If the corresponding IMR bit is set and the Global Interrupt Bit is set, an interrupt will be generated. Instead of overflowing to 00, the counter is automatically reloaded on each overflow.

For example, if the reload value is 0FAh, the counter will count as follows:

0FAh, 0FBh, 0FCh, 0FDh, 0FEh, 0FFh, 0FAh, 0FBh etc.

The overflow pulse is generated during the period *after* the counter value was 0FFh.

Refer to the following registers for more information about using timers and counters:

T1R: Timer 1 Reload Register (page 33)
T1V: Timer 1 Counter Register (page 33**)**
T2R: Timer 2 Reload Register (page 34)
T2V: Timer 2 Counter Register (page 34)

10. Analog Outputs

The RSC-164 offers two separate options for analog output. The DAC (Digital to Analog Converter) output provides a general-purpose 10-bit analog output that may be used for speech output (with the inclusion of an audio amplifier), or that may be used for other purposes requiring an analog waveform. Many speech applications may require only driving a small speaker, however, and cost can be saved in these applications by using the Pulse Width Modulator (PWM) outputs of the RSC-164 instead of the DAC output.

DAC Output

The digital-to-analog converter (DAC) provides its output through DACOUT, with an output impedance of 22KÙ. V_{DACOUT} can swing from 0V to V_{DD}. A conventional audio amplifier and optional volume control may be used to drive a speaker. Filtering above 5 kHz is recommended to provide the correct frequency response.

PWM Output

The two PWM outputs are designed for driving a 32 Ohm speaker at audio frequencies. These signals produce good quality audio with no additional components. These outputs are actually digital outputs that produce a series of high frequency pulses at varying rates that, when filtered by the mechanical dynamics of a speaker, create the effect of a continuously varying analog signal.

Although it is called a pulse *width* modulator, the RSC-164 actually incorporates a pulse *count* modulator: a programmable number of pulses is produced during each sample period. The two PWM outputs connect directly to the two speaker terminals. During operation, the first of the two signals will be at ground and the second will have a pulse train. The pulses will cause the speaker cone to push out (or pull in, depending on the wiring connections). If there are many pulses in a sample period, the speaker will push out a large amount; if there are few pulses, the

speaker will push out just a little. Switching the pulse train to the first output and holding the second output at ground effectively reverses the polarity of the signals, so now the speaker will pull in. By controlling the number of pulses in each sample period and the output on which they appear, the speaker can be made to move in and out as required to reproduce an audio waveform.

The PWM0 pin is shared with the BUFOUT signal, and the PWM1 pin is shared with the -TE signal. At power-on, these pins are configured for non-PWM operation. The BUFOUT and -TE signals are both test signals that are typically not used during normal operation. When the Pulse Width Modulator is enabled via software, the pins are switched over to the PWM function. See page 41 for information about programming the PWM. (Typically the PWM is controlled by Sensory library code, so there is little need for application programs to alter it.)

11. Hardware Debug Features

Special debugging hardware has been incorporated into the RSC-164 to assist developers in producing code quickly. The specialized circuitry provides the following features:

- A 16-bit break register that holds one ROM breakpoint address
- A TRAP bit in the Flags register (Address 0FFh) that enables or disables the breakpoint.
- A vector at 0FFF8h to which the processor is directed when the Program Counter equals the breakpoint address and the TRAP bit is enabled. The break occurs *before* execution of the instruction at the breakpoint address.
- A special page of Code Space (0FF00h-0FFFFh) for holding the resident debugger software. When this page is entered *via a break*, timers and interrupts are suspend. When execution resumes outside this page, timers and interrupts are restored. Entering this page by means other than a break has no special effect, and timers and interrupts continue to operate as normal.
- A vector at 0FFFCh to which the processor is directed if the -TE pin is held low at power-on. This provides a means for starting up in the debugger.

The debug circuitry, in conjunction with debug monitor software resident in the last page of internal ROM, allows examining the contents of Register, Code, and Data Space addresses. Register space contents may be modified, as may Data or External Code space addresses if implemented in a RSC-writeable device.

The breakpoint feature is not aware of ROM bank switching, so any break will go to the break vector of the currently selected ROM Code Space. For this reason the resident debug monitor must be duplicated in each external ROM bank.

Additionally, the RSC-164 supports a 4-byte, bi-directional communication interface using external hardware registers. This interface provides a means for another computer to interact with the resident debug monitor to achieve a full-featured, real-time debugger. Sensory provides such a debugger with its in-circuit emulator provided with the *RSC Development Kit.* The in-circuit emulator offers the ability to dump or patch memory, dis-assemble code, single-step, and run until a break occurs. The in-circuit emulator use and operation is described in the RSC Development Kit Manual.

12. Design Considerations

Speech recognition accuracy can be degraded by a number of factors. A common problem that causes accuracy degradation is noise: both electrical noise within the system and audio noise picked up by the microphone. A major consideration on RSC Designs is the audio preamp circuit. The signal from a typical electret microphone is of the order of one millivolt, and an overall preamp gain of about 2500 is needed to make this signal useable by the RSC. Careful design and checkout are necessary to assure a low-noise system that will give best recognition. Good grounding practice and elimination of crosstalk into the analog circuitry will further help ensure good recognition accuracy. Product design that encourages the user to speak loudly and close to the microphone helps attain a good signal-to-noise ratio.

Analog Design

The RSC-164 contains an analog to digital converter (ADC) with two inputs, AIN0 and AIN1. The low gain input, AIN0, accepts a signal in the range GND to VDD/2 for A/D conversion. The high gain input, AIN1, accepts a signal in the same range, but its peak-to-peak amplitude should be set to 8 times AIN0 using a buffer preamp as shown in block diagram form in Figure 6. The block diagram also illustrates two bits for automatic gain control of the external analog circuitry controlled by 2 I/O pins. AGC control should only be eliminated if these pins are needed for other uses. As described on page 11 in the section on analog outputs, either the DAC output or the Pulse Width Modulation output can be used to drive the speaker.

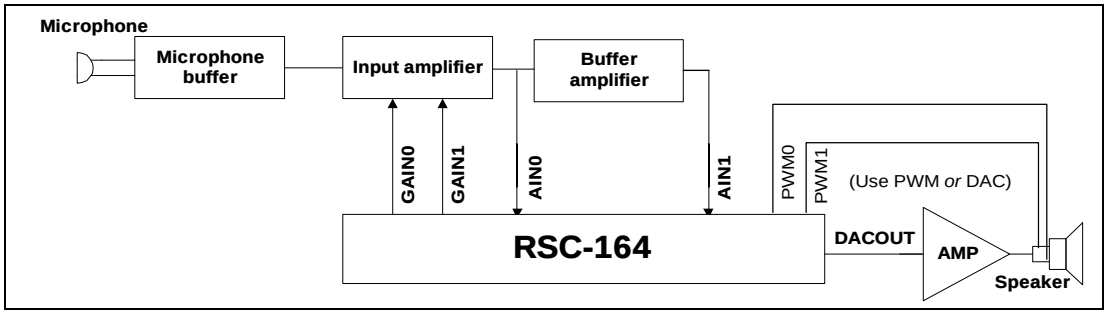


Figure 6 -- External analog Block Diagram

Design of an Input Audio Preamp

This section describes in detail the schematic design and layout considerations for an audio input preamp incorporating 2-bit AGC (Automatic Gain Control).

Table **1** lists the parts count for the analog input audio preamp.

Component	Qty AGC	Component	Qty AGC
Non-Electrolytic Caps (20%)	5	Resistors (5%)	18
Non-Electrolytic Caps (5%)	2	Resistors (1%)	4
Electrolytic Caps	4	LM324 Op Amp	1

Table 1 -- Parts Count for Input Audio Preamp Design

The schematic in Figure 7 illustrates a reference input audio preamp design for use with the RSC-164. Note that all power and grounds shown here are *analog*. See below for power and ground considerations in PCB layout.

Sensory recommends separate analog and digital power and ground supplies for best performance.

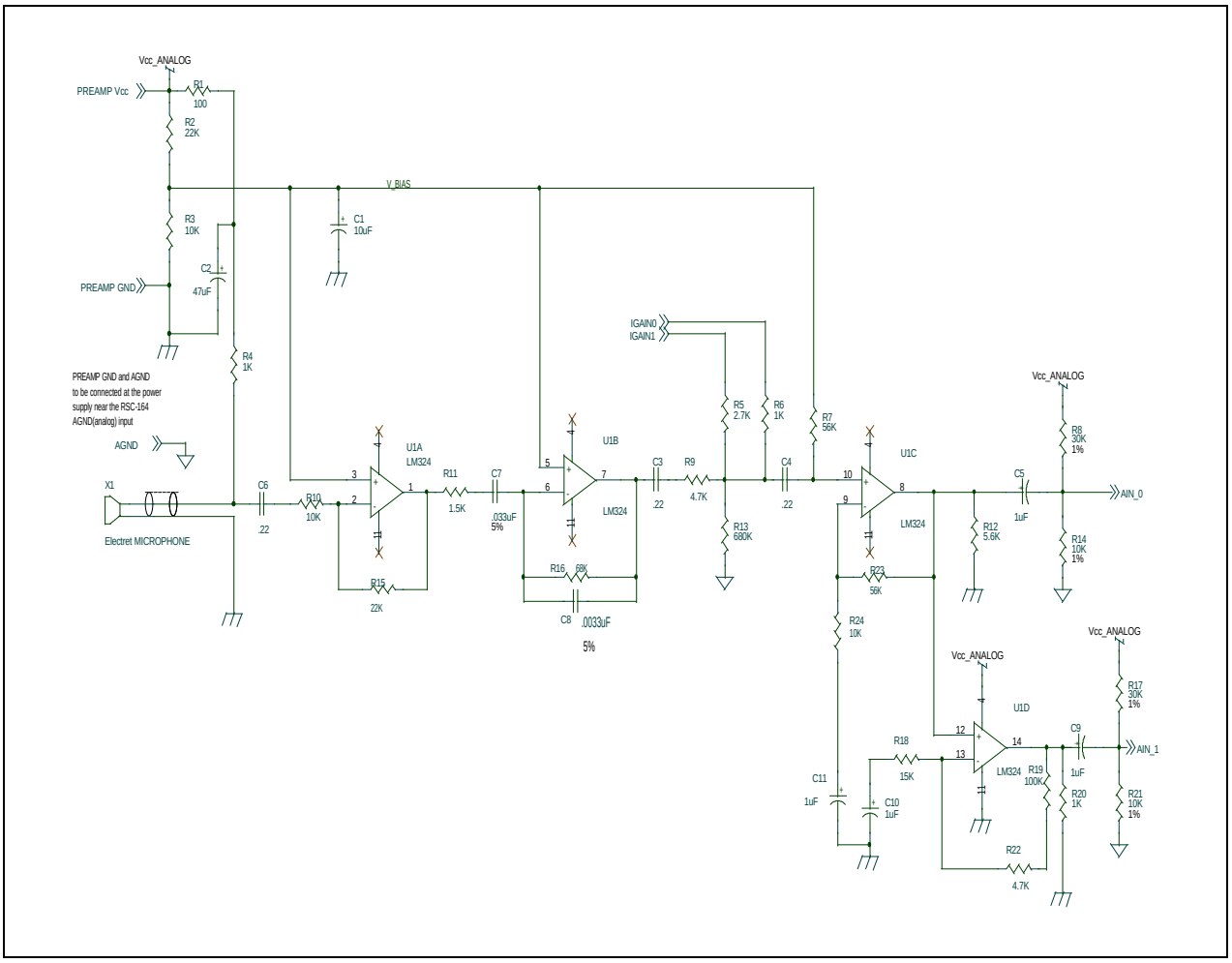


Figure 7 -- Input Audio Preamp with AGC

In order to reduce audio clicks and thumps, it is recommended that audio electronics remain powered as long as the system is turned on. If power is switched on and off actively, the pass transistor should have a high gain and low $V_{CEsat.}$ Although the RSC-164 operates from a wide voltage range, use of external ICs (for instance, ROMs, flash memory, and output power preamps) reduces the voltage range the system can operate in.

PCB Design

A double-sided printed circuit board (PCB) with ground plane should be used. The ground plane should cover all analog circuitry area and only be tied to digital ground near the RSC device. A star ground system should be employed to insure that this single point is the only connection between the digital and analog grounds. Care should be taken to make sure that the analog ground does not carry any digital path current. In order to reduce crosstalk, the analog and digital circuits should be physically separated as far as is practical.

A 0.1 µF bypass capacitor should be installed immediately next to each digital IC and near the V_{DD} pins of the RSC chip. The bypass capacitors should be stacked or monolithic ceramic type, rated at 50 volts. If a three-terminal voltage regulator (such as a 7805) is used, tantalum bypass capacitors should be connected close to the regulator between the input/output pins and ground.

All V_{DD} paths should be derived from a common V_{DD}, but laid out separately in a star pattern with the node near analog V_{DD} of the RSC device. A large electrolytic capacitor should be connected close to the same node. The analog V_{DD} path should be as close to the power input as possible, and care should be taken that this path does not carry digital path current. An example is shown on page 15. The V_{DD} and ground traces between the devices and power supply should be as wide as possible; the main power traces should be no less than 0.080" wide, while power traces to each chip should be no less than 0.060" wide. Using 1.5 to 2.0-ounce copper clad will provide sufficient trace height. Traces for logic signals should be 0.018" to 0.020" wide.

The Data lines should be as wide as possible to reduce impedances. The bi-directional nature of data lines causes large instantaneous currents to get switched around. With inductive loading, these currents can cause data lines to ring and generate large switching spikes. Although these spikes and the ringing will have dampened when the data is sampled, they could damage others devices connected to the same bus. Traces for the data line should be routed so that their length is minimal. These traces should be at least 0.012" wide.

The address, strobe and control signals are outputs of the RSC-164 and have a nominal capacitance; their trace widths and length aren't critical, with a preferred width of more than 0.005". These traces can be made around the power traces and the data line traces.

Shown in figure 8 is a typical implementation of the V_{DD} distribution and grounding ideas mentioned above. The design uses three separate power supply lines: one for the power to RSC core and other digital circuitry (V_{DD}), one for RSC I/O power (V_{DDi}), and one for analog power (AVDD and Preamp V_{CC}). Notice that all power supply lines are buffered from each other by the RC networks.

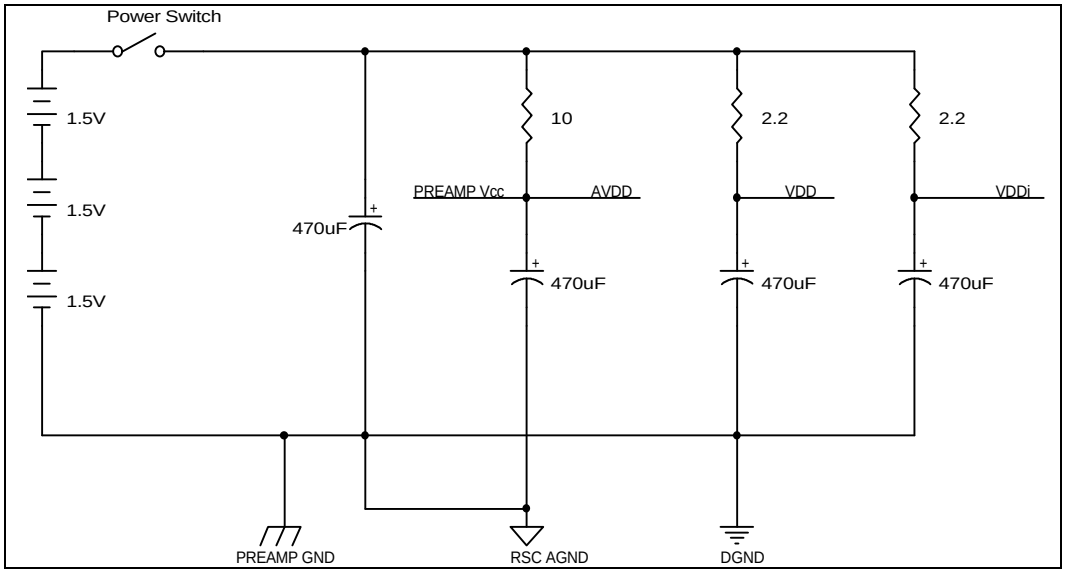


Figure 8 – Power Supply Separation and Grounding

In a practical application using replaceable AA batteries, incorporating a protection diode in series with the power supply will avoid damage to the circuit if batteries are inserted with the wrong polarity.

Locating and Mounting the RSC-164

The RSC-164 may be supplied in packaged form or as a bare die. The die form may be wire bonded directly to the main PCB or, in some cases, may be bonded to a separate chip-on-board (COB) circuit board. In production this COB assembly may be functionally tested, then attached to the main board as a working module.

There are several methods of attaching the COB to the main board, and a careful choice should be made by the designer. The RSC-164 is a 68-pin device requiring good attaching methodology for correct operation. Since cost is always a consideration, COB boards may often be designed as single-sided PCBs.

The simplest way of attaching a single-sided COB to a main board is to lay it, chip side up, on the main board and make solder bridges from the main board up along the thickness of the COB to the electrical contacts on the top of the COB board. In production this is not a reliable technique.

A second technique is to use wires or pins to connect thru-holes between the main board and the COB. This reliable technique may be more time consuming.

A third technique is to put a hole in the main board at the center of the COB location and to mount the COB to the main board UPSIDE DOWN, that is, with the chip facing down into the hole. Then the COB and main boards may be soldered together. In this case, the orientation of the signal leads of the RSC-164 is different from that of the other arrangements.

Figure 9 below shows a suggested parts placement diagram for a double-sided PCB using a die-bonded RSC part with a single RAM or ROM chip. Address and data lines are located along two adjacent sides of the die. Memory and other clock speed digital devices should be located on these sides of the die to avoid having high speed digital signals in the proximity of audio signals. I/O lines may usually be safely routed near analog circuitry, but it is better to keep them away if possible.

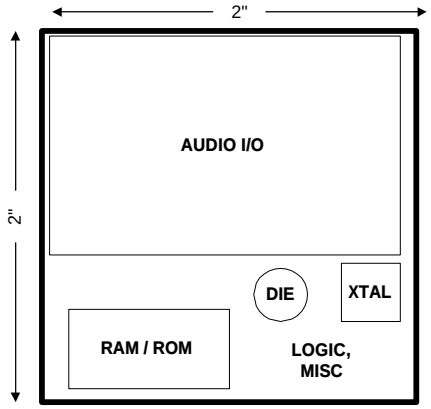


Figure 9 -- Parts Placement with an RSC in Die Form

A packaged RSC part should be placed on a double-sided PCB with RAM or ROM chips as shown in

Figure 10.

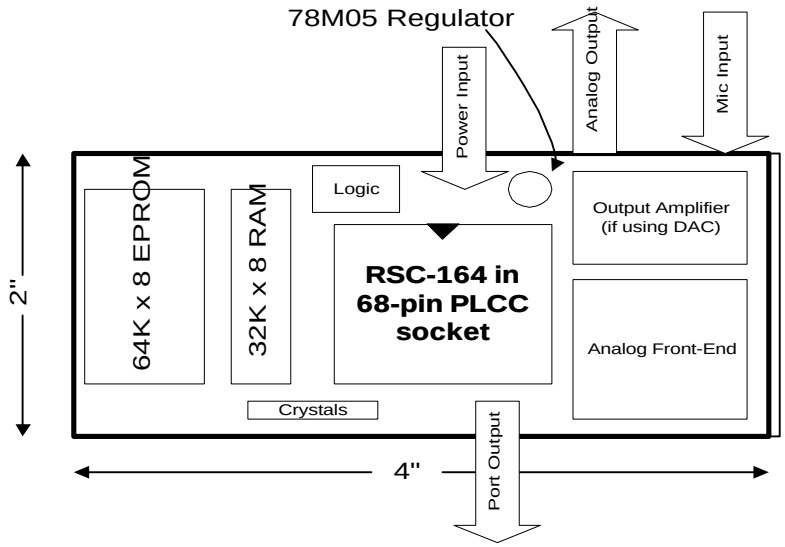


Figure 10 -- Parts Placement with a Packaged RSC with ROM and RAM chips

If the RSC is used in a system with other digital clocks (switching power supplies, LCD driver, etc.) take special care to prevent these signal from getting into the audio circuitry of the RSC.

13. Omni-directional Microphone

Sensory recommends using an omni-directional microphone with -60 dB sensitivity or better. For best performance, speech recognition products should be used in a quiet environment with the speaker's mouth in close proximity to the microphone. If the product is meant to be used in a noisy environment, care should be taken to design around the noise. Improving the signal-to-noise ratio will help make the product a success.

Selecting a Suitable Microphone

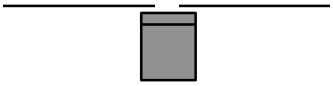
For most applications, an inexpensive omni-directional electret capacitor microphone with a minimum sensitivity of -60 dB is adequate. In some applications, a directional microphone might be more suitable if the signal comes from a different direction than the audio noise. Since directional microphones have a frequency response that depends on their distance from the sound source, such microphones should be used with caution.

Design of Microphone Housing

Proper design of the microphone housing is important, because improper acoustic positioning of the microphone will reduce recognition accuracy. This section describes several important considerations that must be carefully followed in designing the microphone mounting and housing. Many mechanical arrangements are possible for the microphone element, and some will work better than others. We recommend the following guidelines for the microphone housing:

FIRST: In the product, the microphone element should be positioned as close to the mounting surface as possible and should be fully seated in the plastic housing. There must be NO airspace between the microphone element and the housing. Having such an airspace can lead to acoustic resonance, which can affect recognition accuracy.

Good:
Mic. element flush with surface



Bad:
Air cavity between mic. element and surface.

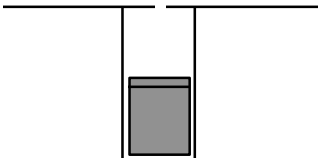


Figure 11 -- Microphone Mounting (1)

SECOND: The area in front of the microphone element must be kept clear of obstructions to avoid interference with recognition. The diameter of the hole in the housing in front of the microphone should be at least 5 mm. Any necessary plastic surface in front of the microphone should be as thin as possible, being no more than 0.7 mm if possible.

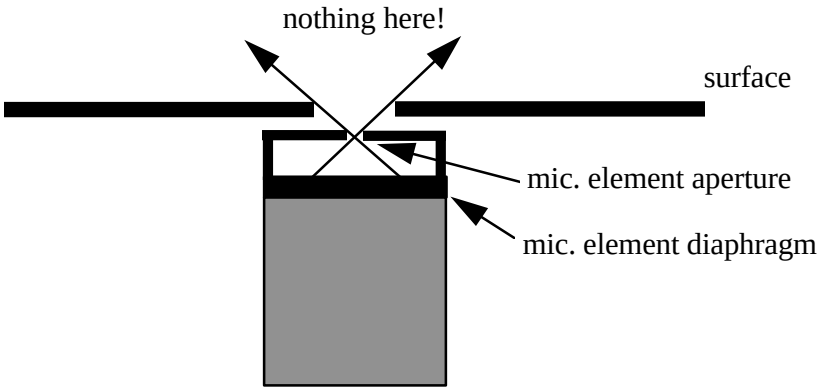


Figure 12 -- Microphone Mounting (2)

THIRD: The microphone should be acoustically isolated from the housing if possible. This can be accomplished by surrounding the microphone element with a spongy material such as rubber or foam. Mounting with a non-hardening adhesive such as RTV is another possibility. The purpose is to prevent auditory noises produced by handling or jarring the product from being “picked up” by the microphone. Such extraneous noises can reduce recognition accuracy.

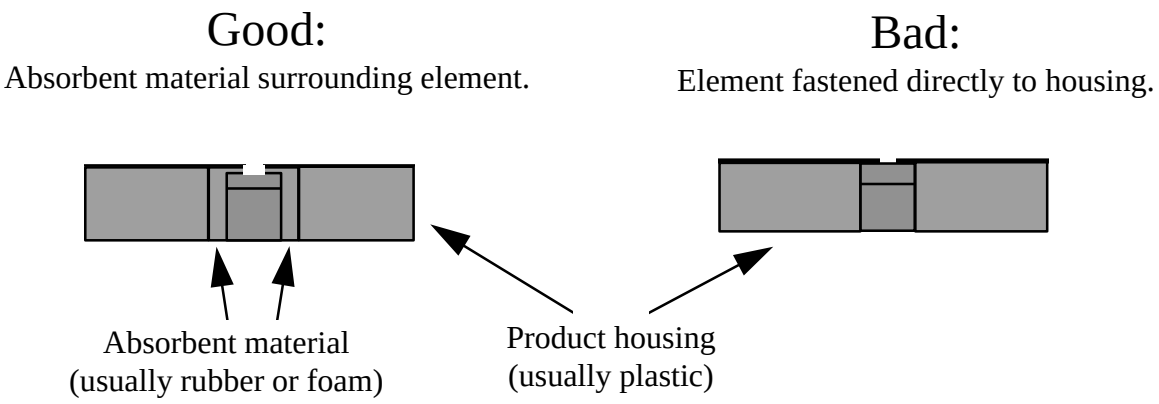


Figure 13 -- Microphone Mounting (3)

If the microphone is moved from 6 inches to 12 inches from the speaker's mouth, the signal power decreases by a factor of four. The difference between a loud and a soft voice can also be more than a factor of four. Thus, the recognizer must function over a wide dynamic range of input signal strength and it will have reduced recognition ability if the input signal either saturates the A/D converter or is too weak. One solution to this problem is to provide Automatic Gain Control (AGC). Sensory-provided software may use I/O lines to provide AGC control from the RSC device to an off-chip analog front-end, which will help compensate for too small or too large a signal. Contact Sensory for the schematic of this AGC audio input amplifier. If the AGC control range is exceeded, software can provide auditory feedback to the speaker about the voice volume. The product can achieve this by saying "please talk louder" or "please don't talk so loud."

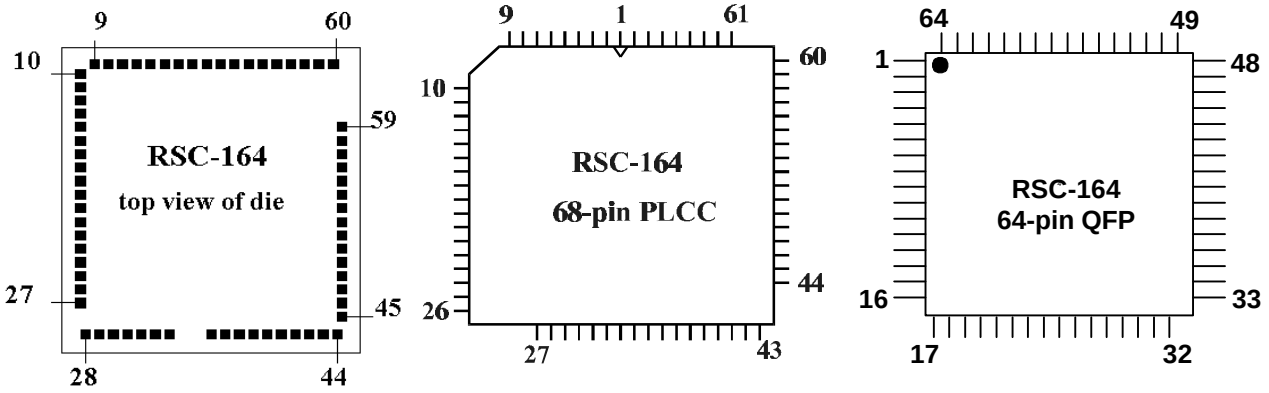
14. Power Consumption and Power Supply Considerations

In operation, the speech recognition circuit may draw a current of around 10 mA. If the system is powered on continuously to listen for a given word, it will drain a button battery in a few hours, or a large alkaline battery in several days. Thus, if the application requires that the recognizer be on all of the time, the system should operate from mains power. Conversely, if the product is designed to operate from batteries, it must usually remain in the low-power "sleep" mode most of the time, until it is occasionally awakened for a few seconds to recognize a word.

When batteries are used, dying batteries will affect performance of the analog front-end, resulting in decreased recognition accuracy. Thus, in order to conserve battery power and to provide good recognition accuracy, if possible, the product should be designed to be activated by the user each time it is required to recognize a word. Current drain is higher during speech or music synthesis because power must be delivered to the speaker.

Mains-powered supply ripple should not exceed 5 millivolts measured between the AVDD and AGND terminals. When using mains power, regulated DC supplies will typically be required. Three-terminal voltage regulators (such as the 7805) are used commonly for this purpose and will provide a minimum of 47 dB of ripple rejection. The power dissipated by these regulators is a function of the voltage differential between the input and output terminals. For instance, powering a 7805 regulator at 9V and delivering 1 amp through it will dissipate 4 watts of power.

15. DIE BOND PAD, PLCC and QFP PIN Descriptions (Version "B")



Name	PLCC Pin/ Die Pad	QFP Pin	Description	I/O
AGND	64	52	Analog Ground. For noise reasons, analog and digital grounds should connect together only at the RSC-164.	-
A[15:0]	10-17, 20-27	1-8, 11-18	External Memory Address Bus	O
AIN0	63	51	Analog In, low gain. (range AGND to AVDD/2.)	I
AIN1	62	50	Analog In, hi gain (8X input amplitude of AIN0, same range)	I
AVDD	67	55	Analog Voltage. For noise reasons, keep this supply independent of digital circuitry.	-
PWM0	65	53	Pulse Width Modulator Output0	O
DACOUT	60	48	Analog Output (unbuffered).	O
D[7:0]	2-9	57-64	External Data Bus	I/O
GND	1, 18, 33, 52	9, 22 41, 56	Digital Ground, CPU core (pins 1 and 33) and I/O (pins 18and 52)	-
PDN	57	NA	Power Down. Active high when powered down.	O
P1[7:0], P0[7:0]	35-42, 43-50	24-31, 32-39	General Purpose Port I/O. Pin P0.0 can act as an external interrupt input. All I/O pins can act as "wake up" inputs.	I/O
-RDC	53	42	External Code Read Strobe	O
-RDD	55	44	External Data Read Strobe	O
-RESET	32	21	Reset	I
SH	61	49	Sample and Hold. Connect a 470 pF capacitor from here to AGND.	I
-TE1/PWM1	66	54	Test Mode *or* Pulse Width Modulator Output1 (multiplexed)	I *or* O
VDD	34, 68	23	Digital Supply Voltage (core)	-
VDDi	19, 51	10, 40	Digital Supply Voltage (I/O line)	-
-WRC	54	43	External Code Write Strobe	O
-WRD	56	45	External Data Write Strobe	O
-XMH	58	46	External Hi-memory enable (low active)	I
-XML	59	47	External Low-memory enable (low active)	I
XO1	30	19	Oscillator 1 output (high frequency)	O
XI1	31	20	Oscillator 1 input	I
XO2	29	NA	Oscillator 2 output (32768 Hz)	O
XI2	28	NA	Oscillator 2 input	I

16. ABSOLUTE MAXIMUM RATINGS

Any pin to GND	-0.1V to +7.5V
Operating temperature (T_O)	0°C to +70°C
Soldering temperature	260°C for 10 sec
Power dissipation	TBD
Operating Conditions	0°C to +70°C;
	V_{DD}=3.5 - 5.0V
	V_{SS}=0V

WARNING: *Stressing the RSC-164 beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

17. D.C. Characteristics

(T_O = 0°C to +70°C, V_{DD} = 5V)

SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS	TEST CONDITIONS
V_{IL}	Input Low Voltage	-0.1		0.75	V	
V_{IH}	Input High Voltage	2.5		Vdd+0.5	V	
V_{OL}	Output Low Voltage		0.3	0.5	V	**I_{OL}**= 4 mA
V_{OH}	Output High Voltage	4.0	4.3		V	**I_{OL}**= -4 mA
I_{IL}	Logical 0 Input Current					
I_{DD1}	Digital Supply Current		10		mA	Osc1 Freq=14.32 MHz, CPU clock divide by 1
I_{DD2}	Analog Supply Current		0.15		mA	Osc1 Freq=14.32 MHz, CPU clock divide by 1
I_{DD3}	Digital Supply Current, Standby					Reserved
I_{DD4}	Analog Supply Current, Standby					Reserved
Rpu	Pull-up resistance P0.0-P1.7	4.5	200	Hi-Z	kΩ	Selected with software

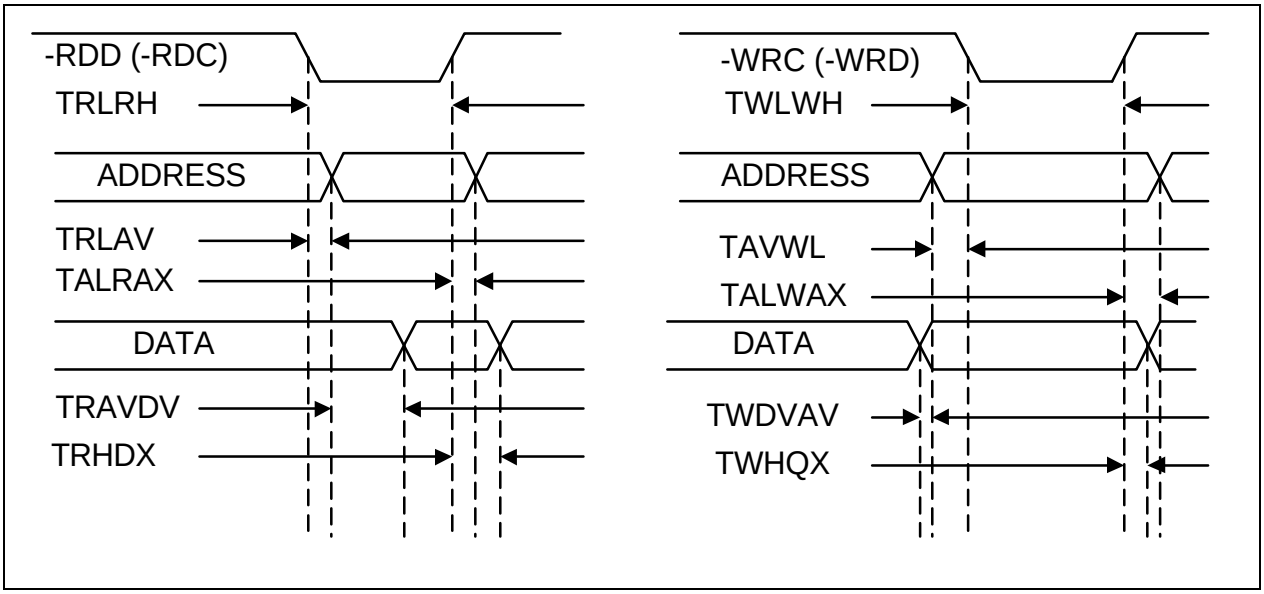
18. A.C. Characteristics *(*External memory accesses*)*

(T_O = 0°C to +70°C, V_{DD} = 5V; load capacitance for outputs = 80 pF; Osc=14.32 MHz)

SYMBOL	PARAMETER	CPU=osc/1, 1 WS MIN	CPU=osc/1, 1 WS MAX	CPU=osc/2, 0WS MIN	CPU=osc/2, 0WS MAX	UNITS
1/TCL1	Processor Clock frequency		14.32		7.16	MHz
TRLRH	-RDC (-RDD) Pulse Width		140		140	ns
TRLAV	-RDC (-RDD) Low to Address valid		5		5	ns
TALRAX	Address hold after -RDC (-RDD)		0		0	ns
TRAVDV	Address valid to Valid Data In		135		135	ns
TRHDX	Data Hold after -RDC (-RDD)	0		0		ns
TWLWH	-WRC (-WRD) Pulse Width		140		140	ns
TAVWL	Address Valid to -WRC (-WRD)	35		70		ns
TALWAX	Address Hold after -WRC (-WRD)	35		70		ns
TWDVAV	Write Data Valid to Address Valid		5		5	ns
TWHQX	Data Hold after -WRC (-WRD)	35		70		ns

Timing Diagrams

Note that the -RDC signal does not necessarily pulse for every read from code space, but may stay low for multiple cycles.



External Read Timing **External Write Timing**

19. RSC-164 INSTRUCTION SET

The RISC instruction set for the RSC-164 has 52 instructions comprising 8 move, 7 rotate, 11 branch, 11 register arithmetic, 9 immediate arithmetic, and 6 miscellaneous instructions. All instructions are 3 bytes or fewer, and no instruction requires more than 8 clock cycles to execute. The column "Cycles" indicates the number of clock cycles required for each instruction when operating with zero wait states. Wait states may be added to lengthen all accesses to external addresses or to the internal ROM (but not internal SRAM). The column "+Cycles/Waitstate" shows the number of additional cycles added for each additional wait state. Opcodes are in HEX.

MOVE Group Instructions

Register-indirect instructions accessing code (*movc*), data (*movx*), or register (*mov*) space locations use an 8-bit operand ("@source" or "@dest") to designate an SRAM register pointer to the 16-bit target address. The "source" or "dest" pointer register must be at an even address. The LOW byte of the target address is contained at the pointer address, and the HIGH byte of the target address is contained at the pointer address+1. The carry, sign, and zero flags are not affected by *mov* instructions.

Instruction	Opcode	Operand 1	Operand 2	Description	Bytes	Cycles	+Cycles/ Waitstate
MOV	10	dest	source	register to register	3	5	3
MOV	11	@dest	source	register to register-indirect	3	5	3
MOV	12	dest	@source	register-indirect to register	3	6	3
MOV	13	dest	#immed	immediate data to register	3	4	3
MOVC	14	dest	@source	code space to register	3	7	4
MOVC	15	@dest	source	register to code space	3	8	4
MOVX	16	dest	@source	data space to register	3	7	4
MOVX	17	@dest	source	register to data space	3	8	4

ROTATE Group Instructions

Rotate group instructions apply only directly to register space SRAM locations. The carry flag is affected by these instructions, but the sign and zero flags are unaffected.

Instruction	Opcode	Operand 1	Operand 2	Description	Bytes	Cycles	+Cycles/ Waitstate
RL	30	dest	-	rotate left, c set from b7	2	5	2
RR	31	dest	-	rotate right, c set from b0	2	5	2
RLC	32	dest	-	rotate left through carry	2	5	2
RRC	33	dest	-	rotate right through carry	2	5	2
SHL	34	dest	-	shift left, c set from b7, b0=0	2	5	2
SHR	35	dest	-	shift right, c set from b0, b7=0	2	5	2
SAR	36	dest	-	shift right arithmetic, c set from b0, b7 duplicated	2	5	2

BRANCH Group Instructions

The branch instructions use direct address values rather than offsets to define the target address of the branch. This implies that binary code containing branches is not relocatable. However, object code produced by Sensory's assembler contains address references that are resolved at link time, so .OBJ modules *are* relocatable. The indirect jump instruction uses an 8-bit operand ("@dest") to designate an SRAM register pointer to the 16-bit target address. The "dest" pointer register must be at an even address. The LOW byte of the target address is contained at the pointer address, and the HIGH byte of the target address is contained at the pointer address+1.

Instruction	Opcode	Operand 1	Operand 2	Description	Bytes	Cycles	+Cycles/ Waitstate
JC	20	dest low	dest high	jump on carry = 1	3	3	3
JNC	21	dest low	dest high	jump on carry = 0	3	3	3
JZ	22	dest low	dest high	jump on zflag = 1	3	3	3
JNZ	23	dest low	dest high	jump on zflag = 0	3	3	3
JS	24	dest low	dest high	jump on sflag = 1	3	3	3
JNS	25	dest low	dest high	jump on sflag = 0	3	3	3
JMP	26	dest low	dest high	jump unconditional	3	3	3
CALL[1]	27	dest low	dest high	direct subroutine call	3	3	3
RET[1]	28	-	-	return from call	1	2	1
IRET	29	-	-	return from interrupt	1	2	1
JMPR	2A	@dest	-	jump indirect	2	4	2

MISCELLANEOUS Group Instructions

Instruction	Opcode	Operand 1	Operand 2	Description	Bytes	Cycles	+Cycles/ Waitstate
NOP	00	-	-	no operation	1	2	1
CLC	01	-	-	clear carry	1	2	1
STC	02	-	-	set carry	1	2	1
CMC	03	-	-	complement carry	1	2	1
CLI	04	-	-	disable interrupts	1	2	1
STI	05	-	-	enable interrupts	1	2	1

[1] Due to limited stack space, use of CALL and RET instructions is discouraged. Subroutines should be called using the software stack macro "SCALL" (See the *RSC-164 Software Library Reference*).

ARITHMETIC/LOGICAL Group Instructions

Arithmetic and logical group instructions apply only to register space SRAM locations. The results of the instruction are always written directly to the SRAM "dest" register. All but the INCrement and DECrement instructions have both register source and immediate source forms.

In each of the following instructions the sign and zero flags are updated based on the result of the operation. The carry flag is updated by the arithmetic operations (ADD, ADC, SUB, SUBC, CP, INC, DEC) but it is *not* affected by the logical operations (AND, TM, OR, XOR). Note: the carry is set **high** by SUB, CP, SUBC, DEC when a borrow is generated.

Instruction	Opcode	Operand 1	Operand 2	Description	Bytes	Cycles	+Cycles/ Waitstate
AND	40	dest	source	logical and	3	6	3
TM	41	dest	source	like AND, destination unchanged	3	6	3
OR	42	dest	source	logical or	3	6	3
XOR	43	dest	source	exclusive or	3	6	3
SUB	44	dest	source	subtract	3	6	3
CP	45	dest	source	like SUB, destination unchanged	3	6	3
SUBC	46	dest	source	subtract w/carry	3	6	3
ADD	47	dest	source	add	3	6	3
ADC	48	dest	source	add w/carry	3	6	3
INC	49	dest	-	increment	2	5	2
DEC	4A	dest	-	decrement	2	5	2
AND	50	dest	#immed	logical and	3	5	3
TM	51	dest	#immed	like AND, destination unchanged	3	5	3
OR	52	dest	#immed	logical or	3	5	3
XOR	53	dest	#immed	exclusive or	3	5	3
SUB	54	dest	#immed	subtract	3	5	3
CP	55	dest	#immed	like SUB, destination unchanged	3	5	3
SUBC	56	dest	#immed	subtract w/carry	3	5	3
ADD	57	dest	#immed	add	3	5	3
ADC	58	dest	#immed	add w/carry	3	5	3

20. RSC-164 SPECIAL FUNCTION REGISTER (SFR) SUMMARY

This section describes the registers located in addresses 0E0h through 0FFh of the register space. These special function registers (SFRs) are generally used for configuration control and system level functions. In many cases an applications programmer might need to access these registers only to initialize the ports. Since the SFRs are extensively used by Sensory's library functions, thorough understanding is essential before changing the contents of these registers..

The Symbol shown is the name recognized by the assembler for the associated register.

Symbol	Address	Register Name	Type	See Page
P0OUT	0E0h	Port 0 Output Register	R/W	27
P1OUT	0E1h	Port 1 Output Register	R/W	27
P0IN	0E2h	Port 0 Input Register	Read	28
P1IN	0E3h	Port 1 Input Register	Read	28
P0CTLA	0E4h	Port 0 Control Register A	R/W	29
P1CTLA	0E5h	Port 1 Control Register A	R/W	30
P0CTLB	0E6h	Port 0 Control Register B	R/W	29
P1CTLB	0E7h	Port 1 Control Register B	R/W	30
CKCTL	0E8h	Clock Control Register	R/W	31
WAKE0	0E9h	Reserved	R/W	32
WAKE1	0EAh	Reserved	R/W	32
T1R	0EBh	Timer 1 Reload	R/W	33
T1V	0ECh	Timer 1 Counter	R/W	33
T2R	0EDh	Timer 2 Reload	R/W	34
T2V	0EEh	Timer 2 Counter	R/W	34
ANCTL	0EFh	Analog Control Register	R/W	35
DAC	0FAh	DAC Hold Register	R/W	36
BANK	0FCh	RAM Bank Select Register	R/W	37
IMR	0FDh	Interrupt Mask Register	R/W	38
IRQ	0FEh	Interrupt Request Register	R/W	38
FLAGS	0FFh	Flags Register	R/W	39

Port 0 Output Register (Address 0E0h)

msb			**P0OUT**				lsb
D7	D6	D5	D4	D3	D2	D1	D0

This register is used to write values to Port 0. Values written to this register affect bits that have been configured as outputs. Bits that have been configured as inputs are not affected.

Port 0 is typically configured as a mixed input and output port and used for interactive purposes such as scanning switches, controlling gains, or lighting LEDs.

Bit Description:

P0OUT:	Output bits D0 through D7
Initialization:	All bits cleared to 0 upon reset.
Read Access:	Read output bits from Port 0
Write Access:	Write output bits to Port 0
Also refer to:	P0CTLA, P0CTLB

Port 1 Output Register (Address 0E1h)

msb			**P1OUT**				lsb
D7	D6	D5	D4	D3	D2	D1	D0

This register is used to write values to Port 1. Values written to this register affect bits that have been configured as outputs. Bits that have been configured as inputs are not affected.

Port 1 is typically configured largely as an output port and used for controlling external memory bank selection for expanded memory. In systems using external code space ROM, bits P1.6 and P1.7 are dedicated to controlling the -XML and -XMH signals.

Bit Description:

P1OUT:	Output bits D0 through D7
Initialization:	All bits cleared to 0 upon reset.
Read Access:	Read output bits from Port 1
Write Access:	Write output bits to Port 1
Also refer to:	P1CTLA, P1CTLB

Port 0 Input Register (Address 0E2h)

msb			**P0IN**				lsb
D7	D6	D5	D4	D3	D2	D1	D0

This register is used to read values from Port 0.

Bit Description:

P0IN: Input bits D0 through D7
Initialization:
Read Access: Input bits from Port 0
Write Access:
Also refer to: P0CTLA, P0CTLB

Port 1 Input Register (Address 0E3h)

msb			**P1IN**				lsb
D7	D6	D5	D4	D3	D2	D1	D0

This register is used to read values from Port 1.

Bit Description:

P1IN: Input bits D0 through D7
Initialization:
Read Access: Input bits from Port 1
Write Access:
Also refer to: P1CTLA, P1CTLB

Port 0 Control Register A (Address 0E4h)

msb			**P0CTLA**				lsb
D7	D6	D5	D4	D3	D2	D1	D0

This register is used with P0CTLB to control the function of the general purpose port 0.

Bit Description:

P0CTLA:
Initialization: All bits cleared to 0 upon reset.
Read Access:
Write Access:
Also refer to: P0CTLB

Port 0 Control Register B (Address 0E6h)

msb			**P0CTLB**				lsb
D7	D6	D5	D4	D3	D2	D1	D0

This register is used with P0CTLA to control the function of the general purpose port 0.

Bit Description:

P0CTLB:
Initialization: All bits cleared to 0 upon reset.
Read Access:
Write Access:
Also refer to: P0CTLA

The control registers A and B together control the function of the general purpose port 0:

B	A	Function
0	0	Input - Weak Pull-up
0	1	Input - Strong Pull-up
1	0	Input - No pull-up
1	1	Output

For example, if register P0CTLB bit 4 is set high, and register P0CTLA bit 4 is low, then pin P0.4 is an input without a pull-up device.

Port 1 Control Register A (Address 0E5h)

msb			**P1CTLA**				lsb
D7	D6	D5	D4	D3	D2	D1	D0

This register is used with P1CTLB to control the function of the general purpose port 1.

Bit Description:

P1CTLA:
Initialization: All bits cleared to 0 upon reset.
Read Access:
Write Access:
Also refer to: P1CTLB

Port 1 Control Register B (Address 0E7h)

msb			**P1CTLB**				lsb
D7	D6	D5	D4	D3	D2	D1	D0

This register is used with P1CTLA to control the function of the general purpose port 1.

Bit Description:

P1CTLB:
Initialization: All bits cleared to 0 upon reset.
Read Access:
Write Access:
Also refer to: P1CTLA

The control registers A and B together control the function of the general purpose port 1:

B	A	Function
0	0	Input - Weak Pull-up
0	1	Input - Strong Pull-up
1	0	Input - No pull-up
1	1	Output

For example, if register P1CTLB bit 3 is cleared, and register P1CTLA bit 3 is set high, then pin P1.3 is an input with a strong pull-up.

Clock Control Register (Address 0E8h)

msb			**CKCTL**				lsb
PD	T2	FC	CD1	CD0	PCS	DO2	EO1

This register is used to enable the two oscillators, select the processor clock source and the internal clock divider, and select timer 2 overflow for some reserved functions. Sensory's library code may initialize specific settings for this register, so it should be changed only with care.

Bit Description:

CKCTL.0: EO1
0: Enable oscillator #1 inverter
1: Disable oscillator #1
Cleared by reset or (reserved event & (PCS=0))

CKCTL.1: DO2
0: Disable oscillator #2 inverter
1: Enable oscillator #2 inverter
Cleared by reset.

CKCTL.2: PCS
0: Processor clock source = oscillator 1
1: Processor clock source = oscillator 2
Cleared by reset.

CKCTL.4-CKCTL.3: CD1-CD0
Select processor clock divisor. The processor clock rate is the fraction of the source clock shown.

CD1	CD0	Division
0	0	1/2
0	1	1/1
1	0	1/8
1	1	½56

Cleared by reset.

CKCTL.5: FC
0: Disable reserved function clock
1: Enable reserved function clock
Cleared by reset.

CKCTL 6: T2
0: Timer #2 overflow does not cause reserved function
1: Timer #2 overflow causes reserved function
Cleared by reset.

CKCTL.7: PD
0: Not reserved funtion
1: Reserved function flag.
Cleared by reset and reserved functions.

Port 0 Reserved Function Configuration Register (Address 0E9h)

msb			**WAKE0**				lsb
D7	D6	D5	D4	D3	D2	D1	D0

Reserved function.

Port 1 Reserved Function Configuration Register (Address 0EAh)

msb			**WAKE1**				lsb
D7	D6	D5	D4	D3	D2	D1	D0

Reserved function.

Timer 1 Reload Register (Address 0EBh)

msb			**T1R**				lsb
D7	D6	D5	D4	D3	D2	D1	D0

This register contains an 8-bit reload value for timer 1. The reload register is readable and writeable by the processor. When the timer overflows from FF to 00, a pulse is generated that sets IRQ #0 (timer #1).

Bit Description:

T1R:
Initialization: All bits cleared to 0 upon reset.
Read Access: Timer #1 Counter Reload (2's complement of period)
Write Access: Timer #1 Counter Reload (2's complement of period)
Also refer to: T1V

Timer 1 Counter Register (Address 0ECh)

msb			**T1V**				lsb
D7	D6	D5	D4	D3	D2	D1	D0

Timer 1 counter is a read-only register. If the processor writes to the counter, the data is ignored, and the counter is preset to the reload register value from T1R. Instead of overflowing to 00, the counter is automatically reloaded on each overflow.

For example, if the reload value is 0FAh, the counter will count as follows:

0FAh, 0FBh, 0FCh, 0FDh, 0FEh, 0FFh, 0FAh, 0FBh etc.

The overflow pulse is generated during the period *after* the counter value was 0FFh.

The input clock for Timer 1 is always generated from oscillator #1, gated by the wake-up delay, gated by bit 7 of the Clock Control Register, CKCTL.7 flag = 0, then divided by 16. For normal operation with a 14.32 MHz crystal, Timer 1 counts at a rate of 0.895 MHz. Thus the longest duration that can be directly timed is 255/(0.895 MHz) = 285 microseconds.

Bit Description:

T1V:
Initialization: All bits cleared to 0 upon reset.
Read Access: Timer #1 current counter value
Write Access: Force asynchronous load of counter from reload register
Also refer to: T1R

Timer 2 Reload Register (Address 0EDh)

msb			**T2R**				lsb
D7	D6	D5	D4	D3	D2	D1	D0

This register contains an 8-bit reload value for timer 2. The reload register is readable and writeable by the processor. When the timer overflows from FF to 00, a pulse is generated that sets IRQ #1 (timer #2).

Bit Description:

T2R:
Initialization: All bits cleared to 0 upon reset.
Read Access: Timer #2 Counter Reload (2's complement of period)
Write Access: Timer #2 Counter Reload (2's complement of period)
Also refer to: T2V

Timer 2 Counter Register (Address 0EEh)

msb			**T2V**				lsb
D7	D6	D5	D4	D3	D2	D1	D0

Timer 2 counter is a read-only register. If the processor writes to the counter, the data is ignored, and the counter is preset to the reload register value from T2R. Instead of overflowing to 00, the counter is automatically reloaded on each overflow.

For example, if the reload value is 0FAh, the counter will count as follows:

0FAh, 0FBh, 0FCh, 0FDh, 0FEh, 0FFh, 0FAh, 0FBh etc.

The overflow pulse is generated during the period *after* the counter value was 0FFh.

The input clock for timer #2 is generated from oscillator #2 divided by 128. With typical operation with a 32,768 Hz crystal for oscillator #2, the count rate for Timer 2 is 256 Hz.

Bit Description:

T2V:
Initialization: All bits cleared to 0 upon reset.
Read Access: Timer #2 current counter value
Write Access: Force asynchronous load of counter from reload register
Also refer to: T2R

Analog Control Register (Address 0EFh)

msb				**ANCTL**			lsb
MD		LS1	LS0	DM	BE	DE	M

The analog control register configures the A/D and D/A. Since the RSC analog signals are normally dedicated to functions associated with Sensory's library code, there is seldom need for applications programs to access this register.

Bit Description:

ANCTL.7: Mode Bit (MD)

If the mode bit is 0, the other bits are as follows:

ANCTL.0: M
1: ADC Mode, comparator powered-up.
0: DAC Mode, comparator powered-down.
Cleared by reset.

ANCTL.1: DE
1: Enable analog output DACOUT
0: Disable analog output DACOUT
Cleared by reset.

ANCTL.2: BE
1: Enable buffered output BUFOUT.
0: Disable buffered output BUFOUT.
Cleared by reset.

ANCTL.3: DM
0: D/A is full-scale.
1: D/A is half-scale.
Cleared by reset.

ANCTL.4: LS0
Provides LSB for full-scale D/A mode.
Cleared by reset.

ANCTL.5: LS1
Provides LSB for half-scale D/A mode, and second to LSB for full-scale D/A mode.
Cleared by reset.

ANCTL.6: Reserved

Analog Control Register (Address 0EFh) (continued)

If the mode bit is 1, the other bits are as follows:

ANCTL.0-ANCTL.1: Control the inverter strength of the 32,768 Hz oscillator

Bit 1	Bit 0	Strength
0	0	5 µA
0	1	10 µA
1	0	20 µA
1	1	40 µA

ANCTL.2-ANCTL.3: Control the output resistor of the 32,768 Hz oscillator

Bit 3	Bit 2	Resistance
0	0	50 KΩ
0	1	100 KΩ
1	0	200 KΩ
1	1	400 KΩ

ANCTL.4-ANCTL.6: Reserved

When reading from the Analog Control Register, the side containing the 32,768 Hz oscillator control parameters is **not** read, while the other side is read.

Read Access:
Write Access:
Initialization: On reset, both sides of the Analog Control Register are set to zero.
Also refer to:

DAC Hold Register (Address 0FAh)

msb			**DAC**				lsb
D7	D6	D5	D4	D3	D2	D1	D0

Holds the eight most significant bits of the value to be converted to an analog signal. Since the analog signal is typically controlled by Sensory library code, there is seldom need for applications programs to access this register. This register is not affected during A/D conversions. It contains a signed, 8-bit number and is cleared to 0 by reset.

Bit Description:

DAC:
Initialization: All bits cleared to 0 upon reset.
Read Access:
Write Access: DAC Hold Value
Also refer to:

RAM Bank Select Register (Address 0FCh)

msb			**BANK**				lsb
W2	W1	W0	B4	B3	B2	B1	B0

The RSC-164 architecture supports 1024 bytes of Register Space (RAM). Only 384 of the maximum 1024 bytes are implemented in the RSC-164. Since the register space instructions support 8-bit addresses, some registers must be addressed through a banking scheme. The bank register generates address bits 9-5 for accesses associated with register space locations 0C0-0DFH, the "map bank". A 10-bit address is always output to the internal SRAM register space (maximum of 1024 bytes), but bits 9-5 are always zero unless the lower bits are in the map bank range.

Bit Description:

BANK.0-BANK.4: These bits map to the specific 32 byte "map bank" (0C0H to 0DFH) any 32-byte block in the internal register space except for the SFR block (the block must be aligned on a 32-byte boundary).

The Special Function Registers (0E0h-0FFh) may only be directly addressed. The first 192 locations (000h-0BFh) may be directly addressed or they may be mapped to the map bank, while the remaining 192 locations (100h-1BFh) may only be accessed via the map bank through bank selection.

Bits B0 through B4 point to the appropriate bank in RAM as follows. Values not shown are prohibited and may produce unexpected results.

Bits B0-B4	**RAM bank mapped to 0C0h-0DFh**
00h	000h-01Fh
01h	020h-03Fh
02h	040h-05Fh
03h	060h-07Fh
04h	080h-09Fh
05h	0A0h-0BFh
08h	100h-11Fh
09h	120h-13Fh
0Ah	140h-15Fh
0Bh	160h-17Fh
0Ch	180h-19Fh
0Dh	1A0h-1BFh
010h or greater	The system will wrap around (SRAM bit9 is ignored in RSC-164)

BANK5-BANK7: WAIT STATES.
These bits define the number of wait states for external/internal memory access (set to 7 on reset). Note that both the internal and external code and data spaces, but not the SRAM, are controlled by these bits. Sensory technology code requires specific wait states, so changes to this register must be restored before invoking any technology code.

BANK:
Initialization: Bits 5-7 set to 1, all other bits cleared to 0 upon reset.
Read Access:
Write Access: Wait State Configuration, Bank selection.
Also refer to:

Interrupt Mask Register (Address 0FDh)

msb			**IMR**				lsb
			E14	EI3	EI2	EI1	EI0

Bit Description:

IMR0-IMR4: EI4:1= enable interrupt request #4 (PWM complete)
EI3:1= enable interrupt request #3 (Positive edge of P00)
EI2:1= enable interrupt request #2 (Reserved)
EI1:1= enable interrupt request #1 (Overflow of Timer 2)
EI0:1= enable interrupt request #0 (Overflow of Timer 1)

IMR5-IMR7: Unused

Initialization: All bits cleared to 0 upon reset.
Read Access:
Write Access: Interrupt Source Selection.
Also refer to: IRQ, FLAGS

Interrupt Request Register (Address 0FEh)

msb			**IRQ**				lsb
			IR4	IR3	IR2	IR1	IR0

Bit Description:

IRQ0-IRQ4: IR4: 1= interrupt request #4 (PWM complete)
IR3: 1= interrupt request #3 (Positive edge of P00)
IR2: 1= interrupt request #2 (Reserved)
IR1: 1= interrupt request #1 (Overflow of Timer 2)
IR0: 1= interrupt request #0 (Overflow of Timer 1)

IRQ5-IRQ7: Unused

Initialization: All bits cleared to 0 upon reset.
Read Access:
Write Access: The bits in this register can *not* be set by writing a one to any bit, but they can be cleared by writing zeroes. To assure that pending interrupts are not lost, a bit should be cleared by using a *mov* instruction, not an *and* instruction. For example, the Interrupt Service Routine for Timer1 should clear the interrupt by the instruction:

```
mov     irq, #~1
```

Do **not** use ‘and’ instructions in the IRQ register.

Also refer to: IMR, FLAGS

Flags Register (Address 0FFh)

msb			**FLAGS**				lsb
C	Z	S	T				GIE

The flags register contains three bits related to the result of the last arithmetic/logical/rotate/misc instruction, one bit that controls breakpoint enabling, and one bit that controls the generation of interrupts. The flags register is not affected by branch instructions, except that an IRET instruction restores the value preceding the interrupt. The flags register is not affected by mov instructions unless it is the destination register.

Bit Description:

FLAGS.0: GIE (Global Interrupt Enable)
0: All interrupts disabled
1: Interrupts Enabled
Cleared by reset

FLAGS.1-FLAGS.3: Reserved

FLAGS.4: T (Trap)
0: Breakpoint function disabled
1: Breakpoint function enabled
Cleared by reset. When the Trap bit is set, the processor will jump to the debug monitor when the Program Counter equals the value in the breakpoint register.

FLAGS.5: S (Sign)
0: Result of last Arithmetic/logical operation was non-negative.
1: Result of last Arithmetic/logical operation was negative
Cleared by reset

FLAGS.6: Z (Zero)
0: Result of last Arithmetic/logical operation was non-zero.
1: Result of last Arithmetic/logical operation was zero.
Cleared by reset

FLAGS.7: C (Carry)
0: No carry from last arithmetic/rotate/misc operation.
1: Last arithmetic/rotate/misc operation produced a carry.
Cleared by reset

21. Special Data Space Addresses Summary

As described previously, the RSC-164 uses *movx* instructions to access all Data Space locations. Typically Data Space locations are external, but a few specific locations are mapped internally in the last page (0FF00h-0FFFFh) of Data Space. For this reason, it is generally best to plan to use *no* external addresses in the last page of Data Space. (The important exception to this is the debugger interface, mapped externally at 0FFFCh-0FFFFh.) Included among the internally-mapped addresses are the Stack registers, the stack pointers, the break register, and the Pulse Width Modulator registers.

Stack Registers (4 each of 16 bits) and Pointers

Occasionally it is useful to manipulate the stack directly (for example, to leave a deeply nested series of calls without unwinding when a fatal error is detected.). Since the RSC-164 stack space is extremely limited, the need for such manipulations is unlikely, but the information here allows doing so if desired.

Address	Location Name	Type	Notes
0FFF0h	Stack 0 Low byte	R/W	
0FFF1h	Stack 0 High byte	R/W	
0FFF2h	Stack 1 Low byte	R/W	
0FFF3h	Stack 1 High byte	R/W	
0FFF4h	Stack 2 Low byte	R/W	
0FFF5h	Stack 2 High byte	R/W	
0FFF6h	Stack 3 Low byte	R/W	
0FFF7h	Stack 3 High byte	R/W	
0FFF8h	Stack WRITE Pointer	R/W	2 LSBs
0FFF9h	Stack READ Pointer	R/W	2 LSBs

Break Registers

The break registers hold the 16-bit break address. These registers may be read or written only when the trap bit is set (see page 39). When the trap bit is set, execution branches to the break vector at location 0FFF8h in Code Space if the Program Counter matches the break address.

Address	Location Name	Type	Notes
0FFFAh	Break Address Low byte	R/W	0FFh on reset
0FFFBh	Break Address High byte	R/W	0FFh on reset

Pulse Width Modulator (PWM) Registers

The Pulse Width Modulator registers enable and control the operation of the PWM. When producing speech or music, these registers are controlled by Sensory's technology code and should not be touched by applications.

Address	Location Name	Type	Page
0FFE0h	PWMCTRL	W/O	41
0FFE1h	PWMA	W/O	41
0FFE2h	PWMDATA	R/W	42

PWM Control Register (Data Space Address 0FFE0h)

msb			**PWMCTRL**				lsb
D7	D6	D5	PWMEN	FADJEN	S2	S1	S0

This register is used to enable the PWM, the PWM frequency adjust, and to select the sample period. The PWM sample rate is derived from Osc#1, divided by various factors controlled by the PWM registers. Sensory's library code may initialize specific settings for this register, so it should be changed only with care.

Bit Description:

PWM_CTRL[2:0]: Sample period
The PWM rate is proportional to 1/(8-S). The fastest rate occurs with S=7. The slowest rate occurs with S=0.
Cleared by reset

PWMCTRL.3: FADJEN
When the FADJEN bit is set, the PWM rate is further reduced by the value in the PWM_A register as described below.
0: Disable PWM Frequency adjust . PWM sample rate = OSC1/(256*(8-S))
1: Enable PWM Frequency adjust. PWM sample rate = OSC1/((512-A)*(8-S))
Cleared by reset.

PWMCTRL.4: PWMEN
0: Disable Pulse Width Modulator outputs
1: Enable Pulse Width Modulator outputs
Cleared by reset.

PWM Adjust Register (Data Space Address 0FFE1h)

msb				**PWMA**			lsb
D7	D6	D5	D4	D3	D2	D1	D0

The PWM Adjust Register contains an 8-bit unsigned value that may further reduce the sample rate of the PWM. If the FADJEN bit is set, then (256-A) wait states are inserted at the end of each sample period.

- If A=0, 256 wait states are inserted.
- If A=0FFh, one wait state is inserted.

During the wait period, the active output goes to zero. Thus, in addition to lowering the sample rate, smaller values of A also reduce the filtered analog output level.

Bit Description:

PWM_A: Frequency adjust bits D0 through D7
Initialization: All bits cleared to 0 upon reset.
Read Access:
Write Access:
Also refer to: PWMCTRL, FLAGS, IMR, IRQ

PWMDATA Register (Data Space Address 0FFE2h)

msb			**PWMDATA**				lsb
D7	D6	D5	D4	D3	D2	D1	D0

The PWMDATA Register contains an 8-bit unsigned value that determines the output pulse pattern (duty cycle) of the active PWM output. When D7=0, the PWM0 output is active and the PWM1 output is zero. The largest output signal for PWM0 is obtained with D=00h. When D7=1, the PWM1 output is active and the PWM0 output is zero. The largest output signal for PWM1 is obtained with D=0FFh. The PWM repeats each data pulse pattern once each sample period. At the end of a sample period, the PWM sets PWMIRQ. The interrupt service routine may provide a new data value. If the IRQ is not serviced, the PWM continues to output the data originally stored in PWMDATA. Output data always lags input by one PWM sample period.

Bit Description:

PWMDATA: Pulse Width Modulator data bits D0 through D7
Initialization: All bits cleared to 0 upon reset.
Read Access:
Write Access:
Also refer to: PWMCTRL, PWM_A

22. Ordering Information

Part	Suffix	Description
RSC-164	none	Unpackaged RSC-164 in die form
RSC-164	P or Q	RSC-164 in 68-pin PLCC package or 64-pin QFP package
RSC-164	C	RSC-164 in 68-pin COB package
RSC-164	i	Unpackaged RSC-164i in die form
RSC-164	iP or iQ	RSC-164i in PLCC package or QFP package